SQL Project: Inventory & Fleet Management for Delivery Service Company

Overview:

You are tasked with developing and querying a database for a delivery service company's Inventory & Fleet Management System. The company needs to manage its delivery fleet (vehicles, drivers), track deliveries, handle warehouse inventory (items, stock), and provide detailed reports for operations and management.

Deliverables:

A fully functioning SQL database with detailed table designs, relationships, and sample data.

SQL queries and scripts for managing the system and generating reports.

Documentation of your approach, optimizations, and decision-making process.

Day 1: Database Design and Basic Operations

Objective:

On Day 1, you'll design the database, create tables with relationships, insert data, and perform basic operations like updating stock levels and tracking deliveries.

Tasks:

1. Database Schema Design

Design the following tables:

Vehicles: Represents delivery vehicles with attributes such as vehicle_id, vehicle_type, capacity, license_plate, and status (available, in-service).

Drivers: Represents the drivers with driver_id, name, license_number, and assigned_vehicle_id (foreign key to Vehicles).

Warehouses: Represents the company's warehouses with warehouse_id, location, and capacity.

Inventory: Represents items in the warehouses with item_id, item_name, stock_quantity, warehouse_id (foreign key to Warehouses).

Deliveries: Represents delivery assignments with delivery_id, vehicle_id, driver_id, warehouse_id, delivery_date, and status (pending, in-progress, completed).

Delivery_Items: Represents items being delivered in each delivery, with delivery_id, item_id, quantity.

2. Data Insertion

Populate each table with at least 5 records:

5 vehicles (different capacities and types).

5 drivers (assign each to a vehicle).

3 warehouses (in different cities).

10 inventory items distributed across the warehouses.

5 deliveries, with at least 3 items per delivery.

3. Basic Data Manipulation

Update stock quantities as items are assigned for delivery.

For each delivery, reduce the stock in the warehouse for the respective items being delivered.

Assign deliveries to drivers and update their vehicle status (in-service while delivering, available after the delivery).

Write queries to:

List all vehicles currently in service.

Get all items that are low in stock (stock quantity < 10).

List all pending deliveries for a specific day.

4. Constraints and Relationships

Add appropriate constraints:

Primary keys for each table.

Foreign keys between related tables (Deliveries and Vehicles, Deliveries and Drivers, Inventory and Warehouses, etc.).

Unique constraint on license plates in the Vehicles table.

5. Prepare Documentation

Document the table schema and the data flow.

Explain the relationships between tables and why you designed them this way.

Day 2: Advanced Queries, Transactions, and Optimization

Objective:

On Day 2, you'll focus on complex reporting queries, error handling with transactions, and performance optimization.

Tasks:

1. Advanced Reporting Queries

Write the following queries:

List all completed deliveries, including the total number of items delivered.

Get a report showing the current stock levels in each warehouse.

Generate a report of all drivers and the number of deliveries they've completed.

Show all deliveries that were in progress or completed within a given time range.

Calculate the total distance covered by each vehicle over the last month (assuming distance data is tracked in the Deliveries table).

Identify any deliveries where the total item quantity exceeds the vehicle's capacity (requires a JOIN between Deliveries, Vehicles, and Delivery_Items).

2. Transactions and Error Handling

Implement a transaction to simulate a delivery process:

Begin a transaction when an item is assigned to a delivery.

Deduct the quantity from the warehouse inventory.

Update the delivery status to in-progress.

If any item's stock is insufficient, rollback the transaction and provide an error message.

If successful, commit the transaction and mark the delivery as completed.

3. Stored Procedures

Create a stored procedure to assign deliveries to drivers and vehicles:

Inputs: delivery_id, vehicle_id, driver_id.

The procedure should:

Check if the vehicle is available.

Assign the vehicle and driver to the delivery.

Update the vehicle status to in-service.

Return a success or failure message based on availability.

4. Performance Optimization

Identify frequently used queries for the reports and add indexes to optimize them. Focus on:

Queries that involve filtering based on date ranges (WHERE clause).

Queries that involve joins between multiple tables (e.g., Deliveries and Delivery_Items).